What is claimed is:

1. A method for mapping procedural code to object-oriented classes, comprising:

starting a graphical user interface ("GUI") in a procedural programming language process space, wherein a user enters a command through the GUI;

initializing a mapping layer in an object-oriented programming language process space, wherein the mapping layer comprises entry-points that have corresponding algorithms that invoke object-oriented class instantiation methods and/or remote method invocations ("RMIs");

executing a GUI callback in response to the command, wherein the GUI callback comprises procedural code and wherein executing a GUI callback in response to the command comprises:

invoking one of the entry-points; and

the mapping layer executing an algorithm corresponding to the invoked entry-point.

2. The method of claim 1, wherein executing the algorithm comprises invoking a class instantiation method.

3. The method of claim 1, wherein executing the algorithm comprises invoking an RMI.

4. The method of claim 1, wherein the procedural programming language is C++.

5. The method of claim 1, wherein the object-oriented programming language is Java.

6. The method of claim 5, wherein the mapping layer is accessed through a Java Native Interface ("JNI") and invoking one of the entry-points comprises:

invoking a JNI application programming interface ("API") call, wherein the JNI API call invokes one of the entry-points.

7. The method of claim 6, wherein the mapping layer is proxied in the procedural programming language process space by a proxy object that includes proxy object methods corresponding to the entry-points, and executing a GUI callback further comprises:

invoking one of the proxy object methods, wherein the invoked proxy object method performs the invoking one of the entry-points step.

8. The method of claim 1, wherein the entry-points are methods of the mapping layer.

9. The method of claim 1, further comprising returning data to the procedural programming language process space.

10. A computer readable medium containing instructions for mapping procedural code to object-oriented classes, by:

starting a graphical user interface ("GUI") in a procedural programming language process space, wherein a user enters a command through the GUI;

initializing a mapping layer in an object-oriented programming language process space, wherein the mapping layer comprises entry-points that have corresponding algorithms that invoke object-oriented class instantiation methods and/or remote method invocations ("RMIs");

executing a GUI callback in response to the command, wherein the GUI callback comprises procedural code and wherein executing a GUI callback in response to the command comprises:

invoking one of the entry-points; and

the mapping layer executing an algorithm corresponding to the invoked entry-point.

11. The computer readable medium of claim 10, wherein executing the algorithm comprises invoking a class instantiation method.

12. The computer readable medium of claim 10, wherein executing the algorithm comprises invoking an RMI.

13. The computer readable medium of claim 10, wherein the procedural programming language is C++.

14. The computer readable medium of claim 10, wherein the object-oriented programming language is Java.

15. The computer readable medium of claim 14, wherein the mapping layer is accessed through a Java Native Interface ("JNI") and invoking one of the entry-points comprises:

invoking a JNI application programming interface ("API") call, wherein the JNI API call invokes one of the entry-points.

16. A computer system that enables the mapping of procedural code to object-oriented classes, comprising:

a memory;

a processor that runs an application, wherein the application generates:

a graphical user interface ("GUI") in a procedural programming language process space, wherein users enter commands through the GUI; and,

a mapping layer in an object-oriented programming language process space, wherein the mapping layer comprises entry-points that have corresponding algorithms that invoke object-oriented class instantiation methods and/or remote method invocations ("RMIs").

17. The computer system of claim 16, wherein the GUI executes callback code in response to an entered command and the executed callback code invokes one of the mapping layer entry-points.

18. The computer system of claim 16, wherein the entry-points are mapping layer methods that are accessed from the procedural programming language process space though application programming interface ("API") calls.

19. The computer system of claim 16, wherein the procedural programming language is C++.

20. The computer system of claim 16, wherein the object-oriented programming language is Java and the object-oriented programming language process space is a Java Virtual Machine.